



# American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Summer 21 22

Section: A

Software Quality Assurance and Testing

## Online Pet Care

A Report submitted By

SN	Student Name	Student ID
1	TASMIA TAMANNA PRIA	18-37746-1
2	SYED RAFIN BIN MOMIN	18-38420-2
3	RAHUL SAHA	18-38439-2
4	ANIK KUMAR SAHA	19-41289-3

Under the supervision of

**ABHIJIT BHOWMIK**

Associate Professor

Department of Computer Science

---

# Software Test Plan

for

<Online Pet Care>

Version 1.0 approved

Prepared by <Pria, Rafin, Rahul, Anik>

<American International University-Bangladesh>

<5 August, 2022>

## Contents

<b>Revision History .....</b>	<b>3</b>
<b>1. TEST PLAN IDENTIFIER: OPC_01 .....</b>	<b>4</b>
<b>2. REFERENCES.....</b>	<b>4</b>
<b>3. INTRODUCTION.....</b>	<b>4</b>
3.1 Background to the Problem.....	4
3.2 Solution to the Problem.....	4
<b>4. REQUEIREMNT SPECIFICATION .....</b>	<b>5</b>
4.1 System Features .....	5
4.2 System Quality Attributes.....	8
4.3 System Interface.....	9
4.4 Project Requirements .....	12
<b>5. FEATURES NOT TO BE TESTED.....</b>	<b>14</b>
<b>6. TESTING APPROACH .....</b>	<b>14</b>
6.1 Testing Levels.....	14
6.2 Test Tools.....	15
6.3 Meetings.....	15
<b>7. TEST CASES/TEST ITEMS .....</b>	<b>15</b>
<b>8. ITEM PASS/FAIL CRITERIA .....</b>	<b>22</b>
<b>9. TEST DELIVERABLES.....</b>	<b>22</b>
<b>10. STAFFING AND TRAINING NEEDS.....</b>	<b>23</b>
<b>11. RESPONSIBILITIES .....</b>	<b>23</b>
<b>12. TESTING SCHEDULE.....</b>	<b>24</b>
<b>13. PLANNING RISKS AND CONTINGENCIES .....</b>	<b>25</b>
<b>14. APROVALS.....</b>	<b>25</b>

## Revision History

Revision	Date	Updated by	Update Comments
0.1	2022.08.05	Anik Kumar Saha	First Draft
0.2	2022.08.08	Tasmia Tamanna Pria	Second Draft
0.3	2022.08.11	Syed Rafin Bin Momin	Third Draft
0.4	2022.08.14	Rahul Saha	Fourth Draft
0.5	2022.08.19	Anik Kumar Saha	Final Draft

## 1. TEST PLAN IDENTIFIER: OPC\_01

## 2. REFERENCES

- Software Requirement Specification (SRS) Document
- <https://mockflow.com/>
- <https://capstoneguide.com/pet-care-management-system-capstone-project-document/>

## 3. INTRODUCTION

### 3.1 Background to the Problem

We who own pets, have to face many problems when it comes to giving our pets immediate aid/treatment. It's a common scenario in every house who owns pets that they can't get their pets immediate treatment because of so many reasons such as being unable to reach vets in time, not being able to know with whom to talk etc. Also, there is another problem and that is, we often encounter street animals getting injured on roads and there also, we are kind of hopeless as by the time help reaches, it's already too late sometimes. And additionally, it is also sometimes difficult to buy foods or toys for our pets from home. So, we decided to make a system by which all these problems can be overcome.

### 3.2 Solution to the Problem

To solve this problem, we have come up with an idea of making a mobile application for easier access to vets and on-time treatment.

Our application is called "PetCare" which will have 4 types of users and they are pet owners, vets, shop owners, rescue center. This app will allow us to do certain operations such as,

- Pet owners can request for pet rescue
- Pet owners can post pictures and videos
- Users can search for pet adoption
- Pet owners can buy foods and toys for their pets
- Shop owner can update products
- Rescue centers can update pet info
- Rescue centers can approve rescue request

- Rescue centers can approve adopting request
- Vets can do consultation with owners who need them

The main goal of this application is to make the above mentioned daily works easier for pet owners as well as for the pets. And the target market for our app are mainly the vets, pet owners, shop owners and rescue centers but in general this app will be helpful for all.

## 4. REQUIREMENT SPECIFICATION

### 4.1 System Features

#### 1. System Registration

Functional Requirements

- 1.1 The software shall allow users to register with the required information
- 1.2 If the username is not unique, the system shall request the user to try registering with different username again.

Priority Level: High

Precondition: Not applicable.

#### 2. System Login

Functional Requirements

- 2.1 The software shall allow users to login with their given username and password
- 2.2 If the username and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.
- 2.3 If the number of login attempt exceed its limit (5 times), the system shall block the user account login for one hour *[optional function]*

Priority Level: High

Precondition: User must have valid user id and password

#### 3. Request for Rescue

Functional Requirements

- 3.1 The software shall allow users to request for rescue, to rescue centers with location and description

3.2 If location is not given, the system must not send request and tell the user to try again with proper location

3.3 Adding description can be optional

Priority Level: High

Precondition: User must have valid account

#### **4. Post Pictures and Videos**

Functional Requirements

4.1 The software shall allow users to post their pets' pictures and videos

4.2 If no post is added, the system must not allow user to submit the post

Priority Level: Low

Precondition: User must have valid account

#### **5. Search for Adoption**

Functional Requirements

5.1 The software shall allow users to search for pet adoption and also adopt pets from the rescue centers

Priority Level: Medium

Precondition: User must have valid account

#### **6. Buy Foods and Toys**

Functional Requirements

6.1 The software shall allow users to purchase toys and foods for their pets from the shop

Priority Level: Low

Precondition: User must have valid account

#### **7. Update Products**

Functional Requirements

7.1 The software shall allow shop owners to update their available products in the app with proper information such as product name, product quantity and product price

7.2 If the required information is not filled, the system must now allow the user to update the product information

Priority Level: Low

Precondition: User must have valid account

## **8. Update Animals' Information**

Functional Requirements

8.1 The software shall allow rescue centers to update the pets' information with proper information through the app so that users can adopt from them

8.2 If the required information is not filled, the system must now allow the user to update the pet's information

Priority Level: Medium

Precondition: User must have valid account

## **9. Approve Rescue Request**

Functional Requirements

9.1 The software shall allow rescue centers to approve rescue request sent by the customers

9.2 A notification must be sent to the customer who have sent the request saying that "We have got your rescue request and we are on our way to reach there."

Priority Level: High

Precondition: User must have valid account

## **10. Approve adopting Request**

Functional Requirements

10.1 The software shall allow rescue centers to approve adopting requests

10.2 A notification must be sent to the customer who have sent the request saying that "We have got your request." With mentioning the time when the pet will be arrived at their doors

Priority Level: Medium

Precondition: User must have valid account

## **11. Start Consultation**

Functional Requirements

11.1 The software shall allow customer and vets to consult when customer wishes to consult

11.2 Customers shall appoint for the specific vets they need and a time will be booked when the vet is available to consult

11.3 Vets shall see which customers are waiting for his response for a consultation and he shall accept as his wish and consult with them after giving them a fixed time by message

Priority Level: High

Precondition: Users must have valid accounts

## 4.2 System Quality Attributes

1. **Usability:** The system shall be effortless to use. The system should be user-friendly and easy to navigate locations.
2. **Efficiency:** The system should make perfect use of the processor capacity and memory. The completion of any task should be completely efficient.
3. **Security:** The system security should be sufficient to prevent unauthorized access to system functions to prevent information loss, to protect the privacy of data and protect the system from virus.
4. **Modularity:** The system's every block of code must be under separate and suitable modules.
5. **Testability:** The system should be easy to test and find defects.
6. **Flexibility:** The system should be flexible enough to be modified.
7. **Reusability:** Code library classes should be generic enough to be reused on different versions of an application or new projects.



### 4.3 System Interface

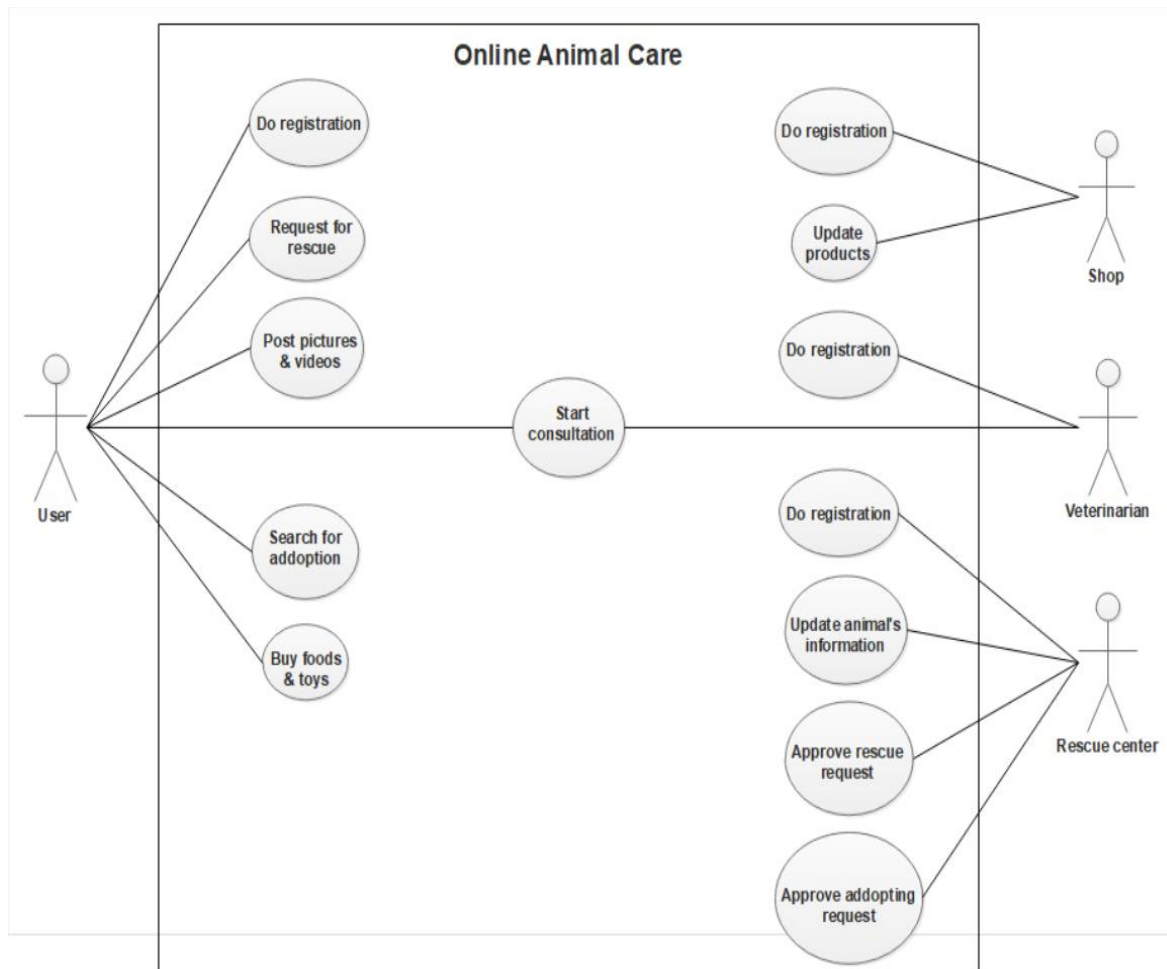


Figure 1: System Overview

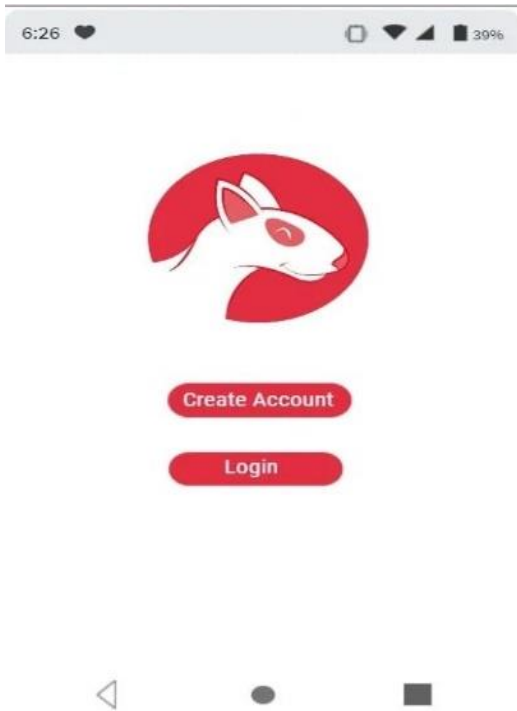


Figure 2: UI for Home page

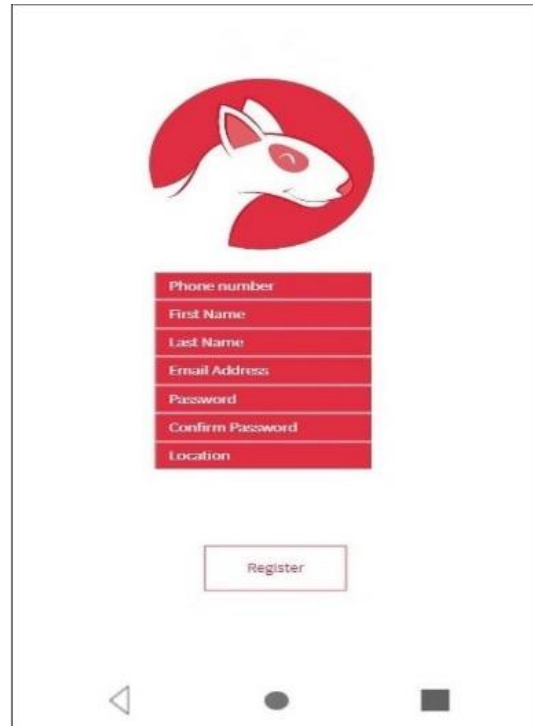


Figure 3: UI for Registration page

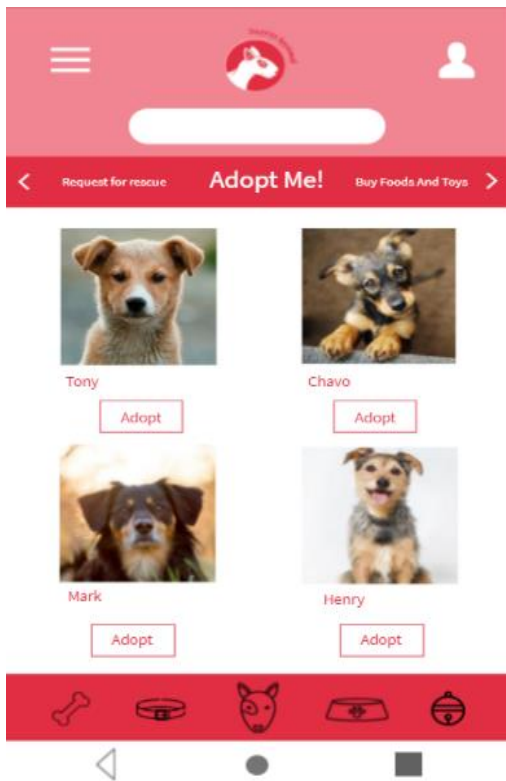


Figure 4: UI for Adopting page

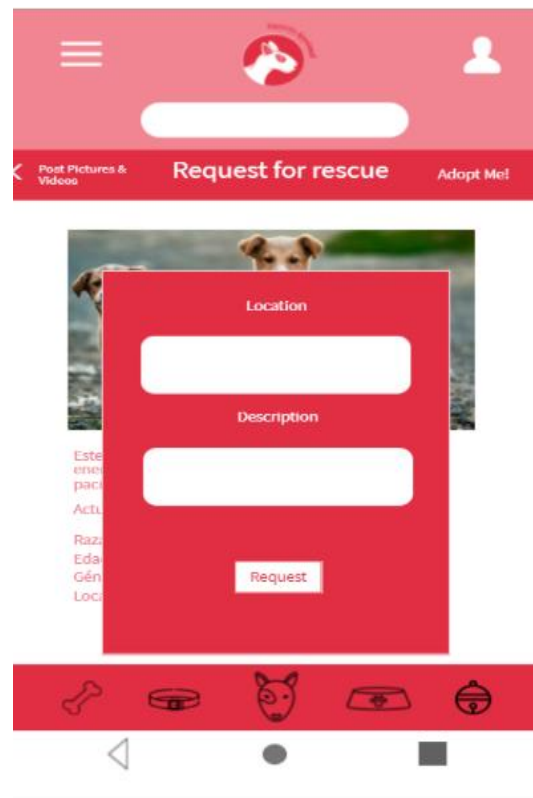


Figure 5: UI for Rescue Request page

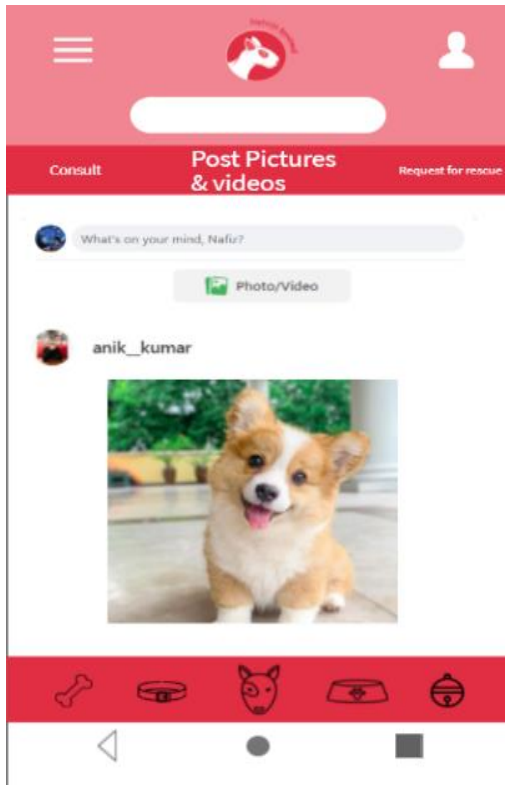


Figure 6: UI for posting page

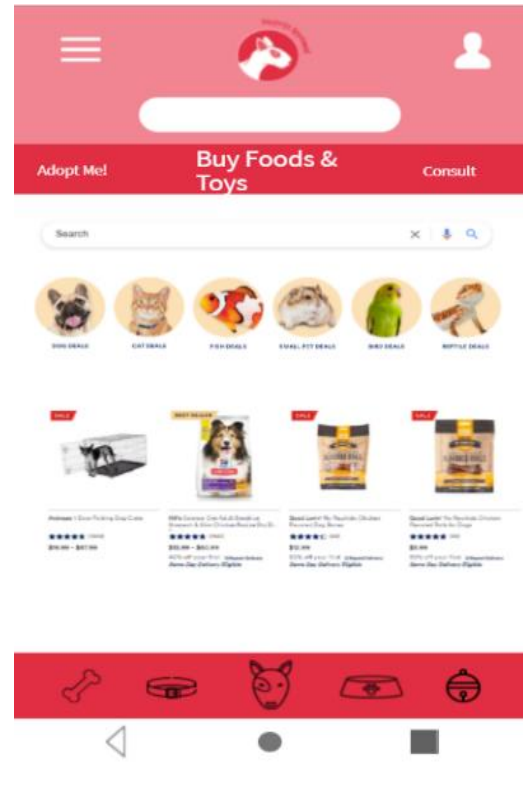


Figure 7: UI for Buying page

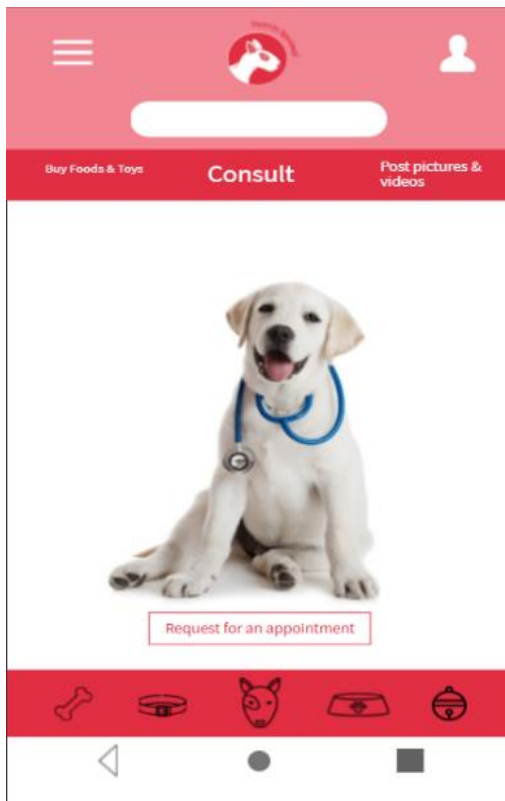


Figure 8: UI for Consulting page (User end)

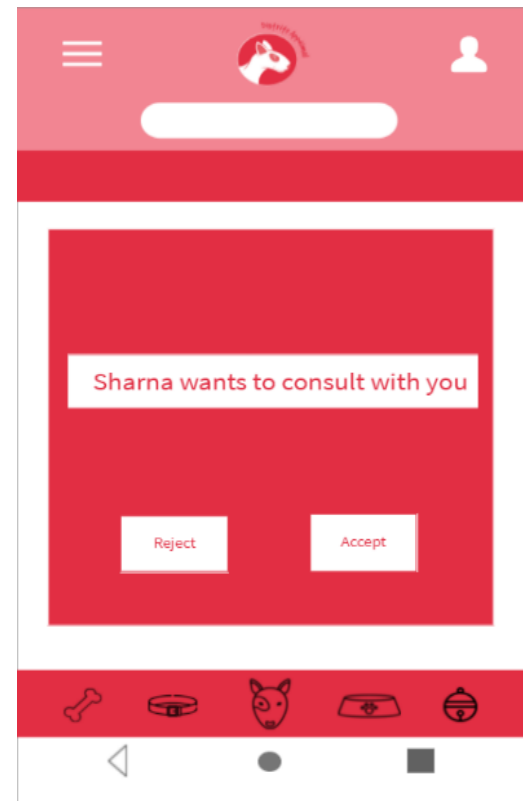


Figure 9: UI for Consultation (Veterinarian end)



Figure 10: UI for Product Update page

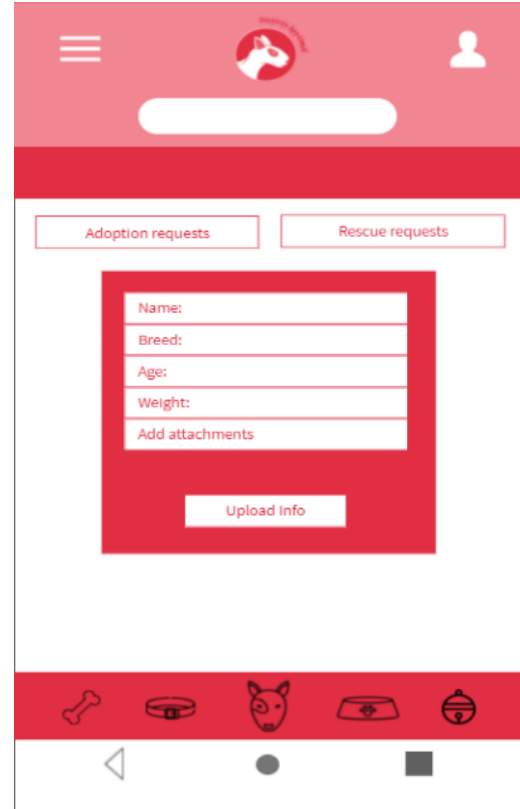


Figure 11: UI for Pets' Info Update page

## 4.4 Project Requirements

1. The code must be written with Dart language.
2. For application shall use Flutter framework.
3. For software databases shall use oracle database server but other databases are also acceptable.
4. For software development, shall use Android studio.
5. For testing, shall use Appium Automation and Flutter Test software.
6. The software size is maximum 200MB.

## Constructive Cost Model

**Software project type:** Organic

**Coefficient**<sub><Effort Factor></sub> = 2.4

So, **P** = 1.05 and **T** = 0.38

**SLOC** = 25000 Lines

**Persons-months, PM** = Coefficient<sub><Effort Factor></sub> \* (SLOC/1000)<sup>P</sup>  
= 2.4 \* (25000/1000)<sup>1.05</sup>  
= 70.48

**Development time, DM** = 2.50 \* (PM)<sup>T</sup>  
= 2.50 \* (70.48)<sup>0.38</sup>  
= 12.59 = 13 months  
= 1560 Working hours in total (Per week 30 hours)

**Required number of people, ST** = PM/DM  
= 70.48/13  
= 5.42 = 6 people

## Budgeting

**Developer/Tester salary of 13 months:**

Per employee salary per month = 35000 Taka = 300 Taka per hour

Total salary = 300 \* 1560 = **4,68,000 Taka**

**Requirement analysis:**

Required time = 1 month = 22 working days = 176 working hour

Requirement analysis person's per hour salary = 200 Taka

Total requirement analysis salary = 200 \* 176 = **35,200 Taka**

**Transportation cost: 10,000 Taka** (Approximate)

**Hardware expense: 1,00,000 Taka** (Approximate)

**Rent expenses:**

Room per month = 12,000 Taka

Total in 12 months = **1,08,000 Taka**

**Total utilities in 13 months: 15,000 Taka** (Approximate)

**Maintenance (Till 3 months after delivery):**

Cost per hour = 1,200 Taka

Total estimated time needed for maintenance = 30 hours

Total estimated maintenance cost = 1,200 \* 30 = **36,000 Taka**

**Project manager's salary of 13 months:**

Per month salary = 30,000 Taka

Total salary =  $30,000 * 13 = 3,90,000$  Taka

**Accountant's salary of 13 months:**

Per month salary = 5,000 Taka

Total salary =  $5,000 * 13 = 65,000$  Taka

**Total expense:**  $4,68,000 + 35,200 + 10,000 + 1,00,000 + 1,08,000 + 15,000 + 36,000$

$+ 3,90,000 + 65,000 = 1,227,200$  Taka

**Profit:** 25% of total expense =  $1,227,200 * 25\% = 3,06,800$  Taka

**Total budget:  $1,227,200 + 3,06,800 = 1,534,000$  Taka**

## 5. FEATURES NOT TO BE TESTED

- Networks
- Hardware
- Users' registration information (Name, Address, Phone number)

## 6. TESTING APPROACH

### 6.1 Testing Levels

The testing for the "PetCare" project will consist of Unit testing, System testing, Integration testing and Acceptance test levels. It is hoped that there will be at least one full time independent test person for system/integration testing. However, with the budget constraints and timeline established; most testing will be done by the test manager with the development teams' participation.

- **Unit testing:** In this stage, every small software unit will be tested time to time by the developer. The code will be reviewed line by line by the programmer by a few techniques. It will basically be a white box testing where no code execution will be done.
- **Integration testing:** A software project consists of multiple software modules, coded by different programmers. In this stage, individual software modules will be combined and tested as a group. This level of testing is to expose defects in the interaction between these software modules when they are integrated. This testing will be done by the tester.
- **System testing:** In this stage, any irregularity will be detected if there is any between the units that are integrated together. This testing defects within both the integrated units and the whole system. This testing will also be done by the tester.

- **Acceptance testing:** In this stage, the testing will determine whether the software system has met the specified requirements or not. It will verify if it has satisfied all the client demands or not. This testing will be done by end-users.

## 6.2 Test Tools

- The tool Testopia will be used for test case management. It is designed to be a generic tool for tracking test cases, allowing for testing organizations to integrate bug reporting with their test case run results.
- The Appium Automation and Flutter Test software will be used to automate the testing.
- Jira project management tool will be used to share documents, communicate with team members, to keep track of schedule and planning, the progress of the testing project and so on.

## 6.3 Meetings

The test team will meet once in every week to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well. These two meetings will be scheduled on different weeks. Additional meetings can be called as required for emergency situations.

## 7. TEST CASES/TEST ITEMS

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Anik		
<b>Test Case ID:</b> PC_01		<b>Test Designed date:</b> 04/08/2022		
<b>Test Priority (Low, Medium, High):</b> High		<b>Test Executed by:</b>		
<b>Module Name:</b> Registration.		<b>Test Execution date:</b>		
<b>Test Title:</b> verify registration with valid information.				
<b>Description:</b> Test the registration page.				
<b>Precondition:</b> User has to provide valid information.				
<b>Dependencies:</b> None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the registration page. 2. Enter required valid	Not Applicable	User should able to registration successfully.	As expected	Pass

information.				
3. Click submit				
<b>Post Condition:</b> After registration user's account is created successfully and all the information is stored in the database, user now can login into the account.				

Project Name: Online Pet Care		Test Designed by: Anik		
Test Case ID: PC_02		Test Designed date: 04/08/2022		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Login		Test Execution date:		
Test Title: verify login with valid username and password				
Description: Test the software login page				
Precondition: user has valid username and password				
Dependencies: Register				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Start the s/w 2. Enter username 3. Enter password 4. Click submit	Username: urs99  Password: 321	User should login into the application	As expected	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				



<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Anik		
<b>Test Case ID:</b> PC_03		<b>Test Designed date:</b> 04/08/2022		
<b>Test Priority (Low, Medium, High):</b> High		<b>Test Executed by:</b>		
<b>Module Name:</b> Request for rescue		<b>Test Execution date:</b>		
<b>Test Title:</b> Verify the requesting page.				
<b>Description:</b> Test the request page.				
<b>Precondition:</b> User has to have a valid account with location				
<b>Dependencies:</b> Location				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the requesting page page. 2. Enter the location and details about the pet. 3. Click request.	Provide valid location.  Provide valid information about the pet.	User should able to request successfully.	As expected	Pass
<b>Post Condition:</b> After requesting the system shows a successful message to the user.				

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Pria		
<b>Test Case ID:</b> PC_04		<b>Test Designed date:</b> 10/08/2022		
<b>Test Priority (Low, Medium, High):</b> Medium		<b>Test Executed by:</b>		
<b>Module Name:</b> Post pictures and videos		<b>Test Execution date:</b>		
<b>Test Title:</b> Verify the posted(timeline) page.				
<b>Description:</b> Test the posted(timeline) page.				
<b>Precondition:</b> User has to be logged in and post with restrictions.				
<b>Dependencies:</b> None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the timeline page. 2. uploads the pictures and videos. 3. Click upload to post	Post the photos and videos with restrictions.	User should able to posted successfully.	As expected	Pass

the photos and videos.				
<b>Post Condition:</b> After posting the system shows a successful message and take back on the timeline to the user.				

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Pria		
<b>Test Case ID:</b> PC_05		<b>Test Designed date:</b> 10/08/2022		
<b>Test Priority (Low, Medium, High):</b> Medium		<b>Test Executed by:</b>		
<b>Module Name:</b> Search for adoption		<b>Test Execution date:</b>		
<b>Test Title:</b> Verify the search for adoption page.				
<b>Description:</b> Test the searching page.				
<b>Precondition:</b> User has to be logged in.				
<b>Dependencies:</b> None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the searching page. 2. Enter required pet to search for adoption. 3. Click Search.	Search with valid data.  Select among the posted pet for adoption.	User should able to search his required pet successfully.	As expected	Pass
<b>Post Condition:</b> After searching the system shows the required pet information to the user.				

<b>Project Name:</b> Online Pet Care			Test Designed by: Rahul	
<b>Test Case ID:</b> PC_06			Test Designed date: 13/08/2022	
<b>Test Priority (Low, Medium, High):</b> Medium			Test Executed by:	
<b>Module Name:</b> Buy foods and toys			Test Execution date:	
<b>Test Title:</b> Verify the shopping page.				
<b>Description:</b> Test the shopping page.				
<b>Precondition:</b> User has to be logged in.				
<b>Dependencies:</b> None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Shopping page. 2. Buy required foods and toys for the pets. 3. Click buy.	Select and payment the products for shopping.	User should able to buy products successfully.	As expected	Pass
<b>Post Condition:</b> After shopping the system will take the user to order status page and the products will deliver within 3/4 days to the user.				

<b>Project Name:</b> Online Pet Care			<b>Test Designed by:</b> Rahul	
<b>Test Case ID:</b> PC_07			<b>Test Designed date:</b> 13/08/2022	
<b>Test Priority (Low, Medium, High):</b> Medium			<b>Test Executed by:</b>	
<b>Module Name:</b> Update Data (Rescue Center)			<b>Test Execution date:</b>	
<b>Test Title:</b> Update animal information				
<b>Description:</b> Update animal information with valid information				
<b>Precondition:</b> User has to be valid and has valid information for update.				
<b>Dependencies:</b> User Login				
<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Status (Pass/Fail)</b>
1. Log in with valid information 2. Click Update	Valid information for update animal	User should be able to update the animal information	As expected	Pass

Information 3. Enter valid information for update 4. Click submit	information			
<b>Post Condition:</b> After Updating the animal information it will redirect to the same show information page and user will be able to see the new added information.				

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Rafin		
<b>Test Case ID:</b> PC_08		<b>Test Designed date:</b> 14/08/2022		
<b>Test Priority (Low, Medium, High):</b> High		<b>Test Executed by:</b>		
<b>Module Name:</b> Approve Request		<b>Test Execution date:</b>		
<b>Test Title:</b> Approve rescue request				
<b>Description:</b> Approve the rescue request from user with information check.				
<b>Precondition:</b> User must have to request for rescue with valid information				
<b>Dependencies:</b> Rescue Location				
<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Status (Pass/Fail)</b>
1. Select rescue request 2. Open request 3. Click approve rescue request	Valid location with rescue request details	User should be able to approve rescue request	As expected	Pass
<b>Post Condition:</b> After approving rescue request user will return to approve request page with other requests.				

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Rafin		
<b>Test Case ID:</b> PC_09		<b>Test Designed date:</b> 14/08/2022		
<b>Test Priority (Low, Medium, High):</b> High		<b>Test Executed by:</b>		
<b>Module Name:</b> Approve Request		<b>Test Execution date:</b>		
<b>Test Title:</b> Approve adopting request				
<b>Description:</b> Approve the animal adopting request from user				
<b>Precondition:</b> User has to chose animal for adoption				
<b>Dependencies:</b> Select Animal				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Select adopting request for animal 2. Open request 3. Click approve adaptation request	Animal adaptation request from user	User should be able to approve animal adaptation request	As expected	Pass
<b>Post Condition:</b> After approving request user will return to approve request page with other requests.				

<b>Project Name:</b> Online Pet Care		<b>Test Designed by:</b> Rafin		
<b>Test Case ID:</b> PC_10		<b>Test Designed date:</b> 13/08/2022		
<b>Test Priority (Low, Medium, High):</b> Medium		<b>Test Executed by:</b>		
<b>Module Name:</b> Update Products (Shop)		<b>Test Execution date:</b>		
<b>Test Title:</b> Update product information				
<b>Description:</b> Update product information with valid information				
<b>Precondition:</b> User has to be valid and has valid information for update.				
<b>Dependencies:</b> User Login				
<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Status (Pass/Fail)</b>
1. Log in with valid information 2. Click Update	Valid information for update Product	User should be able to update the product information	As expected	Pass

Information	information			
3. Enter valid information for update				
4. Click submit				
<b>Post Condition:</b> After Updating the product information it will redirect to the same show information page and user will be able to see the new added information.				

## 8. ITEM PASS/FAIL CRITERIA

The entrance criteria for each phase of testing must be met before the next stage can take place. Now the criteria for pass and fail are given below-

- According to the given scenario, the expected result needs to occur, then the design will be considered pass; otherwise, that criteria should be failed.
- If an item tested ten times, nine times perfectly worked, and a single time does not work correctly, it will be considered as fail case.
- System crash will be considered a failure case.
- After submitting a query in the system, it will be considered a fail case if the expected page doesn't appear.

## 9. TEST DELIVERABLES

- Master test plan (this document)
- Individual test plans for each phase of the testing cycle
- Test Design Specifications
- Acceptance test plan
- System test plan
- Integration test plans
- Unit test plans
- Screen prototypes
- Test scenario and expected result in an excel sheet
- Defect reports and summaries

## 10. STAFFING AND TRAINING NEEDS

This section describes how to approach staffing and preparing the test jobs for the task. It is preferred that there will be at least one full time tester assigned to the project for the system/integration and acceptance testing phases of the project. It is reasonable that the vast majority of the staff will accept some testing jobs. The accompanying jobs are addressed below:

- **Project Manager:** Responsible for the complete execution of the project. This incorporates making necessities, supervising the testing cycle etc. So, project manager must be trained on these fields.
- **Test Manager:** Responsible for making the expert test plans, investigating the test deliverable, dealing with the test cycles and suggesting when testing is finished. So, test manager must be trained on evaluating professional standard test plans.
- **Test Engineer:** Responsible for planning the tests, making the test methods, making the test information, executing tests, composing mechanized test strategies, and detailing measurements to the test administrator. So, test engineers should have the knowledge of planning and executing any test case using automated tools.

## 11. RESPONSIBILITIES

	TM	PM	Dev Team	Test Team	Client
Acceptance test documentation & execution	X	X			
System/Integration test documentation & execution	X	X			X
Unit test documentation & execution	X	X		X	X
System Design Reviews			X		X
Detail Design Reviews			X		X
Test procedures and rules	X	X		X	
Screen & Report prototype reviews			X		X
Change Control and regression testing	X	X	X	X	

## 12. TESTING SCHEDULE

- A. Requirement
- B. Construction prototype
- C. Deliver and feedback
- D. Overall design
- E. Overall design again
- F. Specify module 1
- G. Specify module 2
- H. Specify module 3
- I. Code implementation 1
- J. Code implementation 2
- K. Code implementation 3
- L. Integration testing
- M. System testing

Name/Weeks	1-2	3-4	5-6	7-8	8-9	10-11	12-13	14-15	16-17	18-19	20-21	22-23
A: Anik												
B: Pria												
C: Rafin												
D: Rahul												
E: Pria												
F: Rahul												

[illegible]



### **13. PLANNING RISKS AND CONTINGENCIES**

1. Identify Risks and triggers.
2. Classify and prioritize all risks.
3. Monitor for risk triggers during the project.
4. Might arise time delays during consultation between pet owners and vets.
5. Time delays may occur in navigating location for rescue request.

### **14. APROVALS**

Project Sponsor – Anik Kumar Saha	Approved
Development Management - Tasmia Tamanna Pria	Approved
Edi Project manager - Syed Rafin Bin Momin	Approved
RS Test Manager - Anik Kumar Saha	Approved
Rs Development Team Manager - Syed Rafin Bin Momin	Approved
Reassigned Sales - Tasmia Tamanna Pria	Approved
Order Entry Edi Team Manager - Rahul Saha	Approved

**THE END**